

REMARKS

This amendment is responsive to the Final Office Action dated September 12, 2005. Applicant has amended claims 1-6, 10-12, 17, 18, 20-22 and 24, and cancelled claims 9 and 19. Claims 1-8, 10-18 and 20-25 are pending upon entry of this amendment.

Claim Rejection Under 35 U.S.C. § 103

In the Final Office Action, the Examiner rejected claims 1-3, 5-7, 9, 11-19, 21-25 under 35 U.S.C. 103(a) as being unpatentable over Susai et al. (US 2002/0059428) in view of Sridhar et al. (USPN 6,266,701). The Examiner rejected claims 4 and 10 under 35 U.S.C. 103(a) as being unpatentable over Susai in view of Sridhar and in further view of Bommareddy et al. (USPN 6,779,039). In addition, the Examiner rejected claims 8 and 20 under 35 U.S.C. 103(a) as being unpatentable over Susai in view of Sridhar and in further view of RFC 2616, Fielding et al., 1999.

Applicants respectfully traverse the rejection to the extent such rejections may be considered applicable to the claims as amended. The applied references fail to disclose or suggest the inventions defined by Applicants' amended claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

Applicants have amended the claims to include the requirements that the intermediate device (also referred to as the computer networking device) monitor a plurality of TCP connections (sockets) that connect the device to a server. Based on the monitored response for each of the individual sockets, the intermediate device selects the optimal sockets for forwarding multiplexed HTTP requests to the server. Support for these amendments can be found throughout the present application. For example, pp. 12-13 and FIG. 3 of the present application describe a plurality of agents M1-M6 that are assigned to each client-side socket. Each multiplexing state agent is configured to route requests from the corresponding client to an optimal server-side socket 124b on the networking device for transmission to a server 14 of server system 22. The multiplexing state agent is *free to route subsequent requests from the same client to an optimal server-side socket 124b, whether that be a different server socket, or the same server socket as used for previous requests.* The present application states that, in one embodiment, multiplexor/demultiplexor 80 detects response times for each server socket 124a by

monitoring each server-side socket. The socket having the fastest response time or fewest unfilled requests may be selected.¹ Other parameters may be used to select the particular server socket for multiplexed transmission, such as the last socket accessed, the type of request being made and/or the type of data being requested.²

None of the cited references, either singularly or in combination, teach or suggest an HTTP multiplexor/demultiplexor configured to monitor a plurality of server TCP connections to an individual server, as required by claim 1. Moreover, none of the cited references, either singularly or in combination, teach or suggest an HTTP multiplexor/demultiplexor that includes a plurality of agents, each agent assigned to a different one of the client TCP connections, wherein upon receiving an HTTP request from the client, the respective agent selects one of the plurality server TCP connections based the monitoring of the server TCP connections and routes the HTTP request to the selected server TCP connection for communication to the server, as further required by claim 1.

Sridhar, for example, is entirely silent with respect to monitoring each server TCP connection and then selecting the server TCP connection based on the monitoring. In contrast, Sridhar merely states that multiplexing information “into a single data stream” may provide advantages of higher throughput and lower latency.³ As another example, col. 23, ll. 45-55, of Sridhar only mention that techniques such as multiplexing, protocol spoofing and server site aggregation can be used to reduce latency. In fact, it appears the Examiner recognized the absence of any such teaching in Sridhar stating that “it is implicitly implied by the reference that a single stream takes up less bandwidth than a multiplexed stream.” In other words, the Examiner appears to recognize that, at best, Sridhar merely describe a multiplexed data stream, but fails to provide any teaching or suggestion of mechanisms for actively monitoring a plurality of server sockets from the intermediate device to an individual server and then selectively multiplexing client requests between the different server sockets based on response parameters for the individual sockets.

Like Sridhar, Susai does not provide any teaching or suggestion of a mechanism by which an intermediate device is able to monitor a plurality of server sockets from the

¹ Pg. 14, ln. 16 – pg. 15, ln. 4.

² Pg. 15, ll. 5-12.

³ Abstract.

intermediate device to an individual server, and then multiplex client requests between the different sockets to that same server based on response parameters for each of the sockets. In fact, Susai does not refer to multiplexing between a set of sockets to the same server at all. As correctly recognized by the Examiner, Susai refers to multiplexing across servers. That is, client requests are distributed to different servers (see, e.g., FIG. 4). With respect to communication with a single server, Susai only describes “connection pooling” that the Examiner correctly describes as a technique in which “the connection between interface unit 202 and server S is remained [sic] open for the next incoming stream [to service a subsequent client].” In other words, one client uses a connection until completely finished, and then the system described by Susai reuses the server connection for a subsequent client that initiates a session with that same server.

Thus, neither Sridhar nor Susai teaches or suggests monitoring a plurality of server TCP connections to the server, as required by claim 1. In fact, none of the cited references, either singularly or in combination, teach or suggest an HTTP multiplexor/demultiplexor that includes a plurality of agents, each agent assigned to a different one of the client TCP connections, wherein upon receiving an HTTP request from the client, the respective agent selects one of the plurality server TCP connections based the monitoring of the server TCP connections and routes the HTTP request to the selected server TCP connection for communication to the server over, as further required by claim 1.

Similarly, with respect to claim 3, none of the cited references teach or suggest monitoring a plurality of server TCP connections from a computer networking device to a server to determine a response parameter for each of the server TCP connections. The references fail to teach or suggest monitoring individual sockets to determine a response parameter for *each* socket, as required by claim 3. Moreover, none of the cited references teach or suggest selecting one of the server TCP connections based on the determined response parameter, as required by claim 3. In other words, none of the references relate to selecting between sockets from an intermediate device to a server based on response parameters for the individual sockets. Further, none of the cited references teach or suggest routing the HTTP requests to an individual socket on a server via a multiplexed TCP transmission using the selected server TCP connection.

With respect to independent claim 6, none of the cited references teach or suggest monitoring a plurality of server TCP connections from a computer networking device to a server to determine a response parameter for each of the server TCP connections, and selecting one of the server TCP connections based on the determined response parameter.

With respect to independent claim 17, none of the cited references teach or suggest monitoring a plurality of server TCP connections from the intermediate networking device to the server to determine a response parameter for each of the server TCP connections. Moreover, none of the cited references teach or suggest determining an optimal server socket based on the determined response parameter.

With respect to independent claim 18, none of the cited references teach or suggest an HTTP multiplexor/demultiplexor configured to receive HTTP requests from the clients via a plurality of client TCP connections, and to monitor a plurality of server TCP connections to the server. Moreover, none of the cited references teach or suggest that the HTTP multiplexor/demultiplexor includes a plurality of agents, each agent assigned to a different one of the client TCP connections, and wherein upon receiving an HTTP request from the client, the respective agent selects one of the plurality server TCP connections based the monitoring of the server TCP connections and routes the HTTP request to the selected server TCP connection for communication to the server, the computer networking device being further configured to receive HTTP responses from the server and route the received HTTP responses to a corresponding one of the clients.

With respect to independent claim 22, none of the cited references teach or suggest a computer networking device configured to monitor a plurality of server TCP connections from the computer networking device to a server. Moreover, none of the cited references teach or suggest that the computer network device comprises includes a plurality of agents, each agent assigned to a different one of a plurality of client TCP connections from the computing networking device to the clients. Further, none of the cited references teach or suggest that the agents receive HTTP requests from the clients and distribute those requests via multiplexed transmission over the server TCP connections to a server socket on the server selected based on response parameters determined by monitoring the server TCP connections.

With respect to independent claim 24, none of the cited references teach or suggest a computer networking device configured to monitor a plurality of persistent server socket connections from the computer networking device to a server to determine a response parameter for each of the server TCP connections. Moreover, none of the cited references teach or suggest a device configured to receive HTTP requests from a client, determine an optimal one of the server sockets for each HTTP request based on the respective response parameters for each of the server sockets, and to send each HTTP request to the determined optimal server socket for the request via a multiplexed TCP transmission.

Rejection for Obviousness-type Double Patenting:

The Examiner provisionally rejected claims 1-25 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-26 of copending Application No. 09/882375. Applicants note the provisional status of this rejection. Accordingly, Applicants will address this issue if and when the rejection is formally applied.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date:

November 11, 2005

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

By:

Kent J. Sieffert

Name: Kent J. Sieffert
Reg. No.: 41,312